

Zynq UltraScale+ MPSoC: RPU Introduction Demo Script

Introduction

This demonstration provides high-level instructions on how to run a bare-metal application on an Arm® Cortex®-R5 processor using QEMU command line options.

Preparation:

- Required files: `UED_zcu104.xsa`
The above file can be found in the `$CustEdIP` directory.
- Required hardware: None
- Required software: Vitis™ Unified IDE and QEMU

Zynq UltraScale+ MPSoC: RPU Introduction

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none">• Click the Vitis icon from the taskbar to launch the Vitis Unified IDE.	<ul style="list-style-type: none">• The Workspace Launcher opens after a moment.
<ul style="list-style-type: none">• Close the Welcome tab if it is open.• From the Vitis Components window, click Open Workspace.	<ul style="list-style-type: none">• The Vitis Unified IDE creates a workspace environment that initially only contains a thin structure that tracks tool settings and maintains the log file. As projects are added, this workspace will update to include hardware projects, BSPs, and your software applications.

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> • Select File > New Component > Platform to create a new platform component. • Enter the platform name as rpu_intro_plat. • Ensure that Hardware Design is selected. • Click Browse, navigate to <code>\$TRAINING_PATH/CustEdIP</code>, and select UED_zcu104.xsa. • Click Next. • Select psu_cortexr5_[0 1] as the processor. • Select standalone as the operating system. • Click Next and then click Finish to generate the platform component. 	<ul style="list-style-type: none"> • You are targeting the Zynq™ UltraScale+™ MPSoC platform for the ZCU104 board. Since there is no predefined hardware platform provided in the Vitis Unified IDE, you will need to create a new hardware platform using the provided XSA. • The platform component contains the domain and BSP, along with the FSBL and PMU firmware. <ul style="list-style-type: none"> • The domain contains the BSP for the specific processor for which the application is being targeted.
<ul style="list-style-type: none"> • Select File > New Example > Embedded Software Examples to create a new application component. • Select Hello World from the available examples and click the "+" symbol. • Enter the application name as hello_world. • Click Next. 	<ul style="list-style-type: none"> • This opens the Create Application Component Wizard. • You will find many applications available as examples: <ul style="list-style-type: none"> • DDR Self Refresh • Empty Application • Hello World • Libmetal AMP Demo • lwIP Echo Server • Memory Tests • OpenAMP echo-test • OpenAMP Matrix Multiplication Demo • OpenAMP RPC Demo • Peripheral Tests • Zynq MP FSBL, and many more

Action with Description	Point of Emphasis and Key Takeaway
<ul style="list-style-type: none"> Select rpu_intro_plat as the platform project for this application. Click Next. The domain settings can be left at default, so click Next. Click Finish. 	<ul style="list-style-type: none"> Using the Create Application Component Wizard is a quick way to set up a software application component that targets an existing processor and OS platform (standalone or Linux). You will find two components in the Vitis Components window: <ul style="list-style-type: none"> Platform component (rpu_intro_plat) Application component (hello_world)
<ul style="list-style-type: none"> From the Flow view, select hello_world and click Build. 	<ul style="list-style-type: none"> This will build the platform and application components and generate the ELF file used for running the application.
<ul style="list-style-type: none"> Select Terminal > New Terminal. 	<ul style="list-style-type: none"> This will open the Vitis terminal. QEMU commands are run from here. The current release of the Vitis Unified IDE does not support QEMU. A Tcl and a makefile are provided as a workaround to view the output had the application run on QEMU.
<ul style="list-style-type: none"> Run the following commands to launch QEMU: <pre>[host] \$ cd \$TRAINING_PATH/ rpu_intro/support [host] \$./rpu_intro_demo_ QemuRun.sh r5</pre> This will print: <pre>Hello World</pre> 	<ul style="list-style-type: none"> The files required for running QEMU are provided in the <code>CustEdIP</code> directory. These files need to be copied to the workspace directory for the workaround QEMU to run. Since the real-time processor (RPU) is being targeted, the Arm® Cortex®-R5 processor is chosen.

Summary

You learned how to configure QEMU in the Vitis Unified IDE and saw a bare-metal application run on the Arm Cortex-R5 processor.